

Microsoft Patent Predisclosure Document

Title of Invention: Ink layers on a browser MS # 125345.1

Date: [REDACTED]

Document Author(s): Drew Baird, Suze Woolf

Introduction

[Please provide a high level description of the invention, including the names of the people who contributed to the invention.]

A method of rendering user annotations (such as the underlining, handwritten comments that readers often use as mnemonic aids as well as transparent color areas or "highlighting") as well as removing-"erasing" those annotations on top of web pages.

Motivation for the Invention:

[Describe (1) the problem addressed by the invention (e.g., limitations of prior products of Microsoft, or others), and (2) your solution to the problem (including what "new" things your invention does and a high-level description of how it does them).]

Users need an electronic equivalent of marking, taking notes in or highlighting information they wish to remember in paper books. Since the "pages" of books in electronic books don't exist in physical space, the reader needs a way to attach his annotations to particular locations within the book. Additionally, since the text or graphics on the "page" may be served from a remote server, the annotations are rendered locally in layers on top of the background and text, and do not affect the copy on the server.

Description of the Invention:

[Describe your proposed implementation of the invention, including the architecture and design details of the implementation. The design details should include a description of the component parts of, and individual operations performed by, your implementation. The use of a specific example, showing how the invention solves the problem being addressed, can be particularly helpful. You should also mention whether you have thought of any other implementations, or applications of, your invention. In most cases, 1-2 pages of description should be adequate to start the patent application process, although a more detailed description may greatly enhance the efficiency of the process.]

This invention consists of a method of collecting ink strokes, a method of displaying ink strokes and a method of storing ink strokes.

Ink strokes can be pen, highlight or erase, called "strokes" generically. Ink strokes may have width and color and can be any kind of figure that a user any wish to write on a browser. Highlight strokes are simply ink strokes that have translucent color and usually wider width. Erase strokes are a special case: they are a reconstruction of the original page area under a stroke with a colorless highlight stroke.

Collecting Strokes.

An ActiveX control (ink control) collects strokes via the Mouse Move event in the Windows Operating System and saves them to a file that is directly related to the web page currently being shown. When directed to collect strokes, the ink control will collect strokes while the stylus (or mouse) is down and store them when the stylus (or mouse) is lifted.

Storing Strokes.

Each stroke is saved as it is completed to a file. The file name is determined by the name of the current page. Correlating the ink filename to the page can be accomplished by the browser passing the title in the browser Status Change event, which in turn changes the filename in the control. This is the signal that a new page being drawn. Another method is to wait for the browser DocumentComplete event and read the document name property. As a publishing convention, passing the page name in the Browser Status-Text

Change event is preferred and the actual method implemented. Ink is stored as a file of strokes. A stroke is a set of coordinate pairs of points that define a line. Strokes can also carry attributes with them such as width and color. Highlight strokes also carry an opacity level that describes the translucency of the highlight stroke.

Erase strokes are initially stored and rendered as a special ink. These strokes could be reconciled with the ink it is covering and deleted from the file immediately or at a later time. They could be left in the file to enable an "undo" style. This will give the user option of the feel of a "natural style" erase (remove drawn lines while leaving page text untouched, in a "Paint" application manner) or and "undo style" erase (that is, lines are removed in the reverse order in which they were drawn)

The strokes are stored as a file on the local computer. The browser tracks their exact location. The user can move the file of strokes to another computer and have them rendered essentially the same (or scaled to the current screen size) on any ink-enabled browser.

Displaying Strokes

The ink control will only collect strokes when it is told to collect strokes, that is, when the user has invoked the ink mode. However the strokes must be displayed anytime the browser navigates to a page that has strokes. This necessitates the control operating in the background whenever the user is using the browser. This is accomplished by the ink control looking for any ink file associated with the new page when it encounter each new page. Once it has found that indeed there has been ink saved for this page then it loads it into memory and displays it over the current web page. It also synchronizes with the Draw event of the browser so that any time the browser re-draws its window the ink control re-draws the ink for that page. This is accomplished by the ink control sub-classing (also called hooking) the web browser Draw event and implementing a Callback function to draw the ink. The draw function simply draws all the ink in the file on top of the displayed page from the file.

Diagrams and Flow Charts:

[To support the description provided above, please include: (a) at least one block diagram showing the architecture of the system that implements your invention, and (b) at least one diagram illustrating the primary steps performed by your invention.]

(a) First figure shows ink and highlighting on the actual ink-enabled browser

(b) Second figure shows an architectural view of an ink-enabled browser.

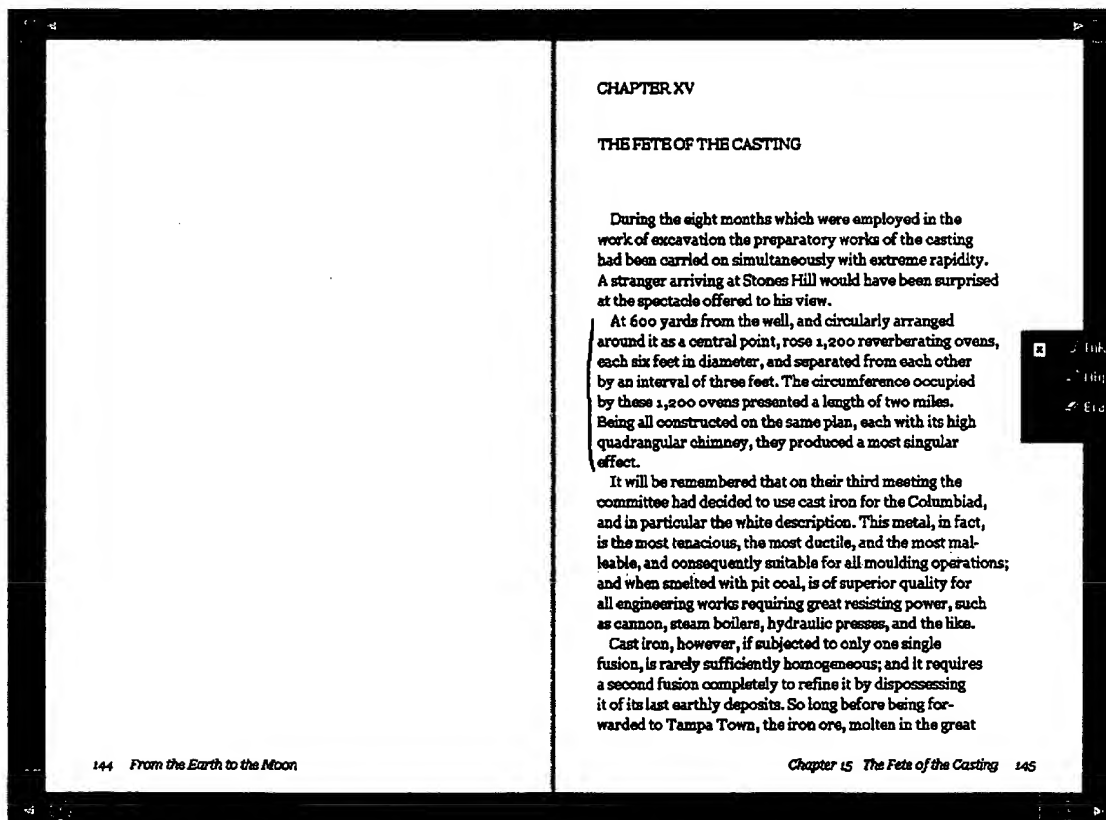
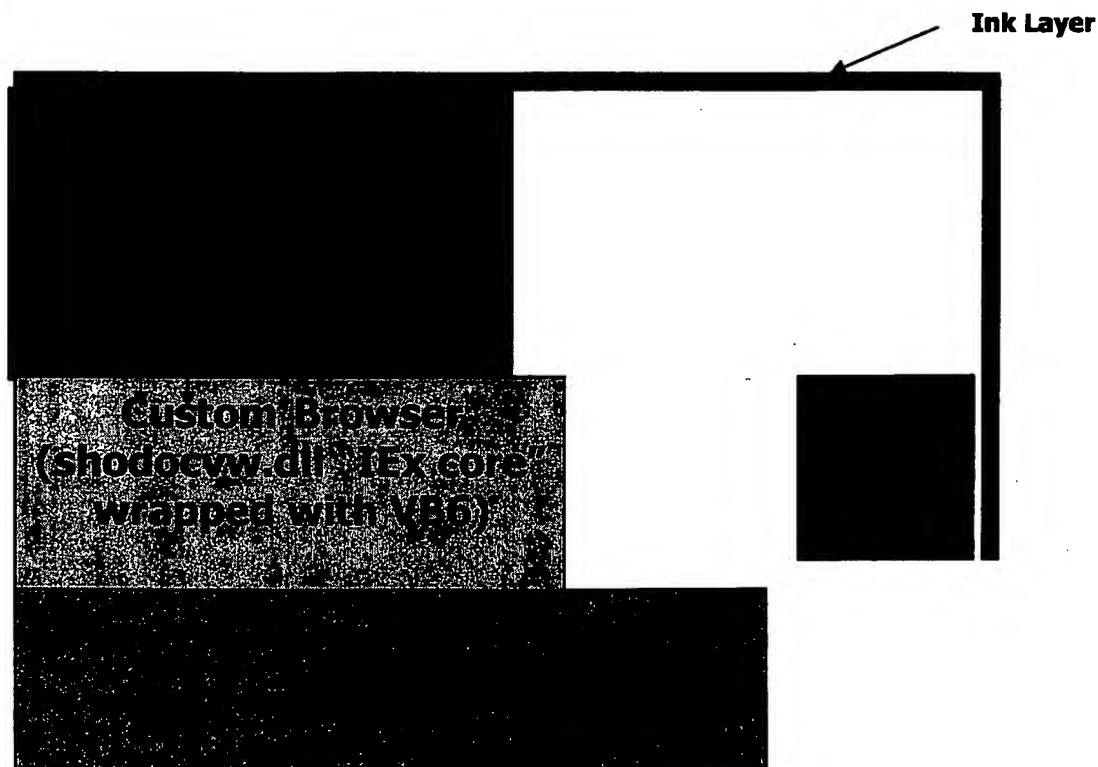


Figure 1.



Additional Information:

- *List the names of any people who contributed to the invention.*

People who contributed to this invention:

Drew Baird
Suze Woolf
John Jenson
John Beezer
Darryl Rubin

(don't know if you need to list what their contributions were...)

- *List any earlier, current or anticipated MS products that may use your invention:*

Anticipate that this "ink on a browser" method may be used in future Microsoft products, especially electronic books and WinCE based products.

- *List and attach (or provide pointers to) any documents that provide additional information about your invention or the product to which it relates, including specifications, journal articles, slide presentations, test/performance results, etc.]*

[click here and type]

- *List any other sources that would provide helpful background information or illustrate prior work of others in this area (including, e.g., journal articles, text books, product literature, products, and specifications):*

[click here and type]